

THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING
ASPECTS OF ELECTRICAL COMMUNICATION

Volume 60

September 1981

Number 7, Part 2

Copyright © 1981 American Telephone and Telegraph Company. Printed in U.S.A.

Digital Signal Processor:

Overview: The Device, Support Facilities, and Applications

By J. R. BODDIE

(Manuscript received February 6, 1981)

This paper introduces the DSP, a new integrated circuit for digital signal processing. We describe the capabilities of the device and the tools available for operating it. Potential applications are also discussed. The paper is an overview of those that follow in this issue of the Bell System Technical Journal.

I. INTRODUCTION

The digital signal processor (DSP) is a new integrated circuit designed by Bell Laboratories and made by Western Electric Company. The device is one of the most complex high-performance circuits developed to date and will have a variety of telecommunications applications. This paper summarizes the capabilities of the DSP, describes user development tools, and lists potential applications.

The Bell System is rapidly applying digital technology to transmission, switching, and station equipment. When signals are encoded digitally, they are easily manipulated by computers and other systems that incorporate advances in very-large-scale integrated circuit technology (VLSI). The VLSI advantages include small size, high reliability, low cost, and low-power consumption. As this trend continues, it is possible to perform previous functions, as well as new ones not possible

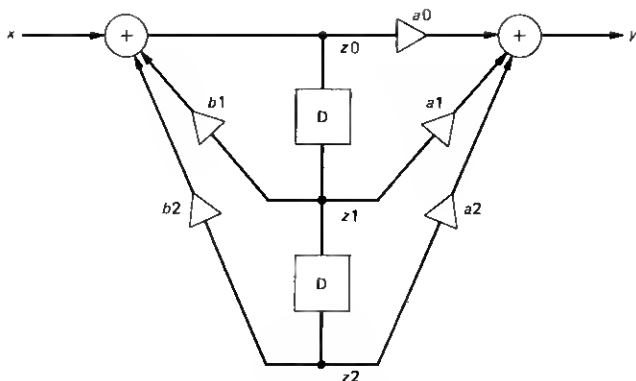


Fig. 1—Second-order filter section.

before, with digital techniques that were formerly performed with analog circuits.

Signal processing is the generation, filtering, detection, and modulation of signals. Most algorithms for signal processing repeatedly use multiplications and additions. A simple example is the second-order section used for filtering. (Refer to Ref. 1 for a more thorough introduction to digital filtering.) Figure 1 is a common schematic representation of the algorithm. The blocks represent delays or storage operations, the triangles are multiplications and the circles are additions. The a_0 , a_1 , a_2 , b_1 , and b_2 in the structure are coefficients that determine the characteristics of the filter. The input, x , is a sequence of numbers representing a continuous waveform. Typically, a new input value is available every $125\ \mu\text{s}$. The output, y , is another sequence of numbers that must be computed using the algorithm at the same rate. The value, z_0 , is an intermediate result, and z_1 and z_2 are delayed values of z_0 . In order to achieve the real-time processing performance required by this example, five multiplications, four additions, and two data movements must be done in $125\ \mu\text{s}$.

The DSP was designed especially for this type of digital signal processing function. Customized by a program in an on-chip, read-only memory (ROM), the device can do over a million high-precision arithmetic computations per second. The key to the performance of the DSP is a parallel, pipelined architecture which provides maximum throughput by keeping all sections of the processor efficiently busy at all times.² The simplified block diagram of the DSP in Fig. 2 shows the organization of the processor as three independently controllable elements: a data arithmetic unit (AU) with multiplier, accumulator, and rounder; an address arithmetic unit (AAU) for controlling memory access; and an I/O unit to provide a serial data interface. A control

unit (CU) synchronizes those elements and provides instruction decoding. Temporary results are stored in a read/write data memory (RAM). For program development and device testing, external memory can be used to replace the on-chip ROM. The DSP can do all the operations required to implement the second-order section of the example in 4 μ s, that is, it can do 31 second-order sections in 125 μ s. This speed is sufficient to implement a complete receiver for *TOUCH-TONE*® telephone service³ or a low-speed modem using a single DSP. The DSP functions in a stand-alone fashion in many applications, but it can easily be interfaced to microprocessors or additional DSPs to achieve a greater degree of signal processing capability.

II. DEVELOPMENT OF THE DSP

The DSP is realized in *N*-channel MOS technology, using depletion loads. Packaged in a 40-pin DIP, it requires only a single 5-volt supply and runs at a 5-MHz rate. The circuit consists of approximately 45,000 transistors within a 68.5-mm² area.

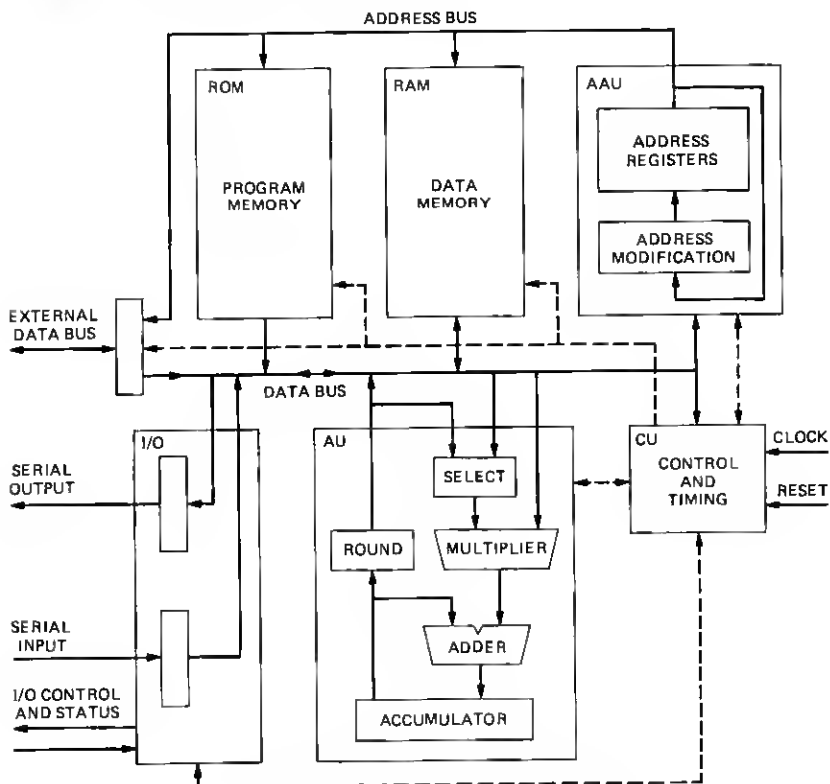


Fig. 2—Digital signal processor block diagram.

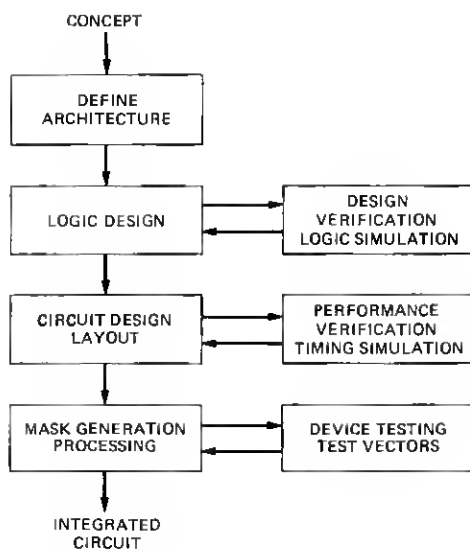


Fig. 3—Device development steps.

This level of integration and the performance requirements presented new challenges for circuit designers in the implementation and testing of the device. Figure 3 shows the major design steps from concept to final product. In the first step, the signal processing requirements of several benchmark applications were combined with the knowledge of what could be done within the limits of the technology. The result was a definition of the architecture, instruction set, and performance specifications.

The logic design work produced a gate-level description of the processor. A logic-level simulator program was used to verify the design. A TTL prototype of the device was also constructed which could emulate the DSP running at full speed. This proved useful for additional design verification and for the development of early DSP applications.

The circuit design and layout implemented the logic design with transistors. A custom layout style was used for the data AU and memories in which each device was created and connected to optimize circuit density, speed, and power. The circuit design and layout of the I/O, AAU, and CU was done using a technique of interconnecting standard predefined logic cells. Custom cells were defined where high-speed paths were required. A computer-aided design system was used to automatically place and connect the cells according to the logic description. This technique greatly reduced the design time of the project. The performance of the design was verified by a circuit simulator program. Computer aids were also used to check for viola-

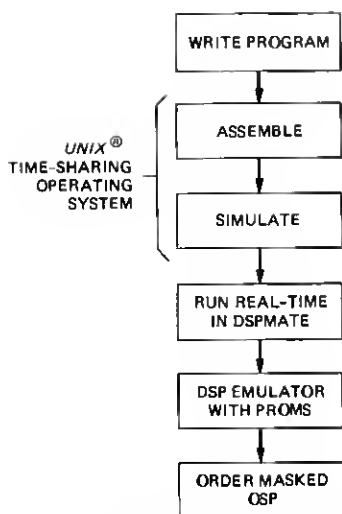


Fig. 4—Digital signal processor application development procedure.

tions of physical layout design rules and to determine the size and type of transistors and parasitic capacitances. In addition, computer plots of the circuit layout were visually inspected for design rule, functional and interconnect errors. Refer to Ref. 4 for details of this design.

Finally, masks were made and devices were fabricated, packaged, and tested. A sequence of inputs designed to test all DSP functions was used to test the devices, as well as to locate as many faults in the chip as possible.⁵

III. SUPPORT TOOLS FOR THE DSP

To facilitate the design of systems using the DSP, a comprehensive set of hardware and software design aids were developed. These tools can be illustrated with a typical application development process.

A DSP system development begins (as shown in Fig. 4) with the writing of a DSP program. The program is entered into a computer by a *UNIX** time-sharing operating system text editor. Digital signal processor programs are written in a unique assembly language which uses standard mathematical notation instead of a more conventional mnemonic format. This greatly improves the readability of programs which are usually arithmetic-intensive because of the nature of signal processing algorithms. An assembler program handles the peculiarities of this pipelined processor in translating DSP programs into a machine-executable code.⁶

* Registered trademark of Bell Laboratories.

The second step in program development is the simulated execution of the program by software simulation of the DSP.⁷ The simulator allows the monitoring of operations in the DSP and the analysis of results at the programmer's computer terminal.

When the user is satisfied with the simulation results, the program can then be run on a DSP in real time. A hardware development system called DSPMATE (Fig. 5) does this while retaining most of the debugging aids of the simulator. The DSPMATE is a microprocessor-based system which can load DSP programs from the computer that hosts the development software. A numeric editor in the system can be used to enter or modify programs and data. The editor can also display a plot of the data on the user terminal. During the execution of a DSP program, I/O events and program flow can be captured and displayed. The DSPMATE can be interfaced to the user's system through a cable and 40-pin plug which is compatible with the DSP. Thus, the operation of the DSP can be monitored in the user's particular hardware environment. Auxiliary circuit boards for the DSPMATE provide analog-to-digital and digital-to-analog conversion, ROM programming, and multiple DSP emulation capabilities.

As program modifications become infrequent, the DSPMATE can be replaced in the user's prototype with a DSP EMULATOR (Fig. 6). This development tool is a small printed wiring board with a DSP and ultraviolet erasable, programmable ROMs for storing the user program.



Fig. 5—Hardware development system (DSPMATE).

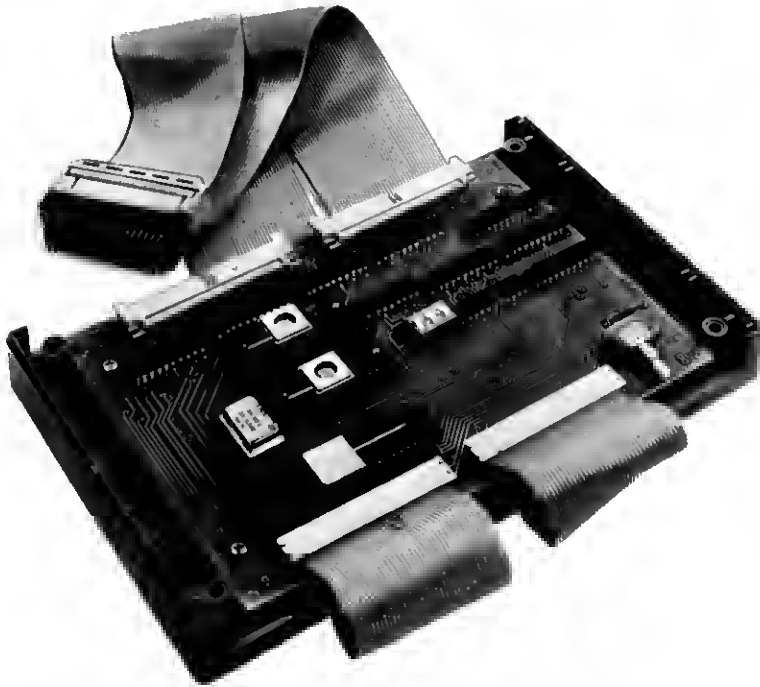


Fig. 6—Digital signal processor EMULATOR prototyping board.

A set of connectors allows the user to monitor the DSP with other test equipment.

Finally, the system developer can request fabrication of a DSP with an on-chip ROM coded with the debugged program.

Extensive documentation on the device, as well as on these tools, and a "hotline" for providing designers with quick answers to their questions, complete the support package.

IV. APPLICATIONS FOR THE DSP

A wide variety of applications exist for the DSP. Table I shows how the device fits into the spectrum of signal processing applications for telecommunications and the number (or fraction) of DSPs required for a particular application.

In this issue, we describe the following applications that represent a small fraction for which the DSP is being considered:

Table I—Application complexity of DSP

Application	Complexity (No. of DSPs)
Second-order section	0.03
ADPCM* coder	0.25
Dual-tone receiver	0.90
Modem, 1200-baud	1.50
Transmultiplexer	6.00

* Adaptive differential pulse-code modulation.

Transmission system measurements

To make measurements on transmission systems, it is useful to be able to generate complex signals comprised of sinusoidal components of arbitrary frequency, phase, and amplitude.⁸ At the other end of the transmission system, loss and noise can be measured by using a high-precision filtering and averaging technique.⁹

Signaling receivers

The detection of tones for signaling systems is an example of applications that must be implemented digitally, because the overall system operates by signals that have been encoded into a digital format.^{3,10}

Line treatment

An application which demonstrates the advantages of the high-precision arithmetic capability of the DSP is the treatment of special voice frequency metallic circuits.¹¹ Here, the DSP provides variable line equalization, gain, and balance functions.

Speech coding

Low bit-rate speech encoding has been made more viable with the DSP. Not only are such systems more economical, but now numerous techniques can be quickly developed and evaluated.^{12,13,14}

Speech synthesis

Man-machine speech communications has long been a subject of intense research activity—usually requiring large general-purpose computers in a nonreal-time mode. The DSP makes real-time, high-quality speech synthesis a practical possibility.¹⁵

V. CONCLUSION

The development of the DSP is a significant accomplishment in many respects. Architecturally, it is a new type of parallel, pipelined processor which achieves a high degree of throughput for the particular

job for which it was designed and, yet, it has the flexibility to do the nonsignal processing tasks when necessary. The circuit is one of the most complex and high-performance devices designed for manufacture. New tools were designed to support the processor and the development of signal processing systems. Many applications that were made with analog elements are now economically feasible through digital techniques, and new algorithms can be explored.

In the future, new DSP architectures will be developed to take advantage of advances in VLSI technology. Thereby, system designers will have the best signal processing components available for applying forward-looking solutions to needs of the Bell System.

VI. ACKNOWLEDGMENT

Many talented people contributed to the development and support of this complex device; however, D. C. Stanzone receives special recognition for his innovation, organization, and direction of the DSP project.

REFERENCES

1. E. J. Angelo, "Digital Signal Processor: A Tutorial Introduction to Digital Filtering," B.S.T.J., this issue.
2. J. R. Boddie et al., "Digital Signal Processor: Architecture and Performance," B.S.T.J., this issue.
3. J. R. Boddie, N. Sachs, and J. Tow, "Digital Signal Processor: Receiver for *TOUCH-TONE*® Service," B.S.T.J., this issue.
4. F. E. Barber et al., "Digital Signal Processor: An Overview of the Silicon Very-Large-Scale-Integration-Implementation," B.S.T.J., this issue.
5. I. I. Eldumiati and R. N. Gadenz, "Digital Signal Processor: Logic and Fault Simulations," B.S.T.J., this issue.
6. C. L. Semmelman, "Digital Signal Processor: Design of an Assembler," B.S.T.J., this issue.
7. J. Aagesen, "Digital Signal Processor: Software Simulator," B.S.T.J., this issue.
8. D. L. Favin, "Digital Signal Processor: Tone Generation," B.S.T.J., this issue.
9. S. P. Cordray, D. L. Favin, and D. P. Yorkgitis, "Digital Signal Processor: Power Measurements," B.S.T.J., this issue.
10. R. N. Gadenz, "Digital Signal Processor: Tone Detection for CCITT No. 5 Transceiver," B.S.T.J., this issue.
11. A. C. Bolling and R. L. Farah, "Digital Signal Processor: Voice-Frequency Transmission for Special-Service Telephone Circuits," B.S.T.J., this issue.
12. D. A. Berkley et al., "Digital Signal Processor: Adaptive Differential Pulse-Code-Modulation Coding," B.S.T.J., this issue.
13. R. E. Crochiere, "Digital Signal Processor: Subband Coding," B.S.T.J., this issue.
14. D. A. Berkley, N. S. Jayant, and C. A. McGonegal, "Digital Signal Processor: Private Communications," B.S.T.J., this issue.
15. M. R. Buric, J. Kohut, and J. P. Olive, "Digital Signal Processor: Speech Synthesis," B.S.T.J., this issue.

